

In the Specification:

Please replace the paragraph beginning on page 1, line 6, with the following rewritten paragraph:

A 1
Conventional methods for interfacing between procedures in a program include those which pass arguments, use an interface area, and use an external file. Herein, the term "interface area" refers to a common block area and an external variable area allocated by using such commands as the "COMMON statement" in FORTRAN and the "extern statement" in C and C++, respectively.

Please replace the paragraph beginning on page 1, line 19, with the following rewritten paragraph:

A 2
However, it has been conventionally assumed that the "COMMON statement" is to be executed in a serial processing, so that one common area has been statically allocated for each executable program unit. As such, there is a possibility of occurrence of a ~~trouble~~problem as follows, when the "COMMON statement" is directly applied to a parallel processing.

Please replace the paragraph beginning on page 2, line 25, with the following rewritten paragraph:

A 3
Herein, the term "interface area" refers to a common block area and external

3
A variable area allocated by using commands such as the "COMMON statement" in FORTRAN and the "extern statement" in C and C++, respectively. Making use of such a common block area or an external variable area, procedures in a program are interfaced with each other.

Please replace the paragraph beginning on page 7, line 12, with the following rewritten paragraph:

4
A When such a rewritten source program is executed, the subroutines XAXIS and YAXIS are called from the main routines in CPU 0 and CPU 1, respectively, as shown in FIG. 8(C). Further, there is considered a situation where there is used variable "a" allocated in a static area for on per program unit by the COMMON statement, as an interface between the main routine and the subroutine XAXIS and YAXIS. In this situation, one variable "a" per program unit is allocated in the static area. As such, the subroutines XAXIS and YAXIS are obliged to use one variable "a" allocated in the common static area. As a result, when a numeric value of the variable "a" is updated after the variable "a" is referred to in the subroutines XAXIS and YAXIS, before the numeric value set to the variable "a" is processed by one of the subroutines, the value may be updated by the other of the subroutines depending on the processing timing of CPU's, leading to a possibility that those operation results in the respective subroutines differ from the values expected by a user. Thus, in the conventional technique, the operation results obtained by the respective subroutines are

rendered to be undefined, so that the source program has not been able to deal~~alt~~deal with a

parallel processing.
